

AMENDMENTS TO THE CLAIMS

For the convenience of the Examiner, all claims have been presented whether or not an amendment has been made.

1. **(Previously Presented)** A method of identifying a plurality of nodes on a network, comprising:

receiving a query sent from a caller node, wherein:

the query comprises a delay constant; and

the query is received by at least one of a plurality of nodes on a network;

determining at the at least one node an answer to the query;

calculating a delay period based at least in part on the delay constant and a network address associated with the at least one node; and

after the delay period, forwarding the answer to the query from the at least one node to the caller node, wherein the caller node is operable to maintain a list of nodes which responded to the query.

2. **(Previously Presented)** A method as recited in claim 1, further comprising:

calculating a timeout period based at least in part on a network address; and

after the timeout period, identifying a plurality of nodes which responded to the query.

3. **(Previously Presented)** A method as recited in claim 1, wherein each of the plurality of nodes on the network forwards to the caller node an answer to the query at different times.

4. **(Previously Presented)** A method as recited in claim 1, wherein each of the plurality of nodes on the network calculates a respective delay period by multiplying the delay constant by its own network address.

5. **(Previously Presented)** A method as recited in claim 1, wherein each of the plurality of nodes on the network are on a subnet, the query sent from the caller node comprising a subnet mask.

6. **(Previously Presented)** A method of identifying nodes on a network, comprising:

sending a query from a caller node to a subnet, the query comprising a delay constant; receiving the query at a plurality of nodes on the subnet;

calculating a delay period based at least in part on the delay constant and a network address associated with at least one of the plurality of nodes;

after the delay period, sending a local response to the query from the at least one node to at least one other node on the subnet;

receiving, at one of the plurality of nodes on the subnet, the local response and compiling a list identifying responding nodes; and

sending the list of responding nodes to the caller node.

7. **(Original)** A method as recited in claim 6, wherein each of the plurality of nodes on the subnet sends its local response at different times.

8. **(Previously Presented)** A method as recited in claim 7, wherein each of the plurality of nodes on the subnet calculates a respective delay period to wait prior to sending its local response.

9. **(Original)** A method as recited in claim 7, wherein the one of the plurality of nodes on the subnet compiling the list identifying the responding nodes is a node first to respond with a local response to the query.

10. **(Original)** A method as recited in claim 9, wherein the node compiling the list identifying the responding nodes calculates a timeout period indicating when a last of the plurality of nodes will send its local response and receives the local responses until the timeout period has expired.

11. **(Previously Presented)** A method as recited in claim 10, wherein the timeout period is calculated by the node compiling the list identifying the responding nodes by multiplying, by the delay constant, an address of a node having a highest IP address on the subnet.

12. **(Canceled)**

13. **(Previously Presented)** A method as recited in claim 11, wherein the query further comprises a subnet mask.

14. **(Previously Presented)** A method as recited in claim 6, wherein the query from the caller node comprises information identifying which of the plurality of nodes on the subnet is to compile the list of responding nodes.

15. **(Original)** A method as recited in claim 14, wherein each of the responding nodes sends its local response to the node identified in the query.

16. **(Previously Presented)** A method of controlling a node in a network, comprising:

receiving at the node a query from a caller node, the query comprising a delay constant;

determining an answer to the query;

calculating a delay period to wait before responding to the query, the delay period based at least in part on the delay constant and a network address associated with the node;

determining whether the node has a lowest address in the network;

if the node does not have the lowest address in the network, waiting the period of time and then responding to the query;

if the node does have the lowest address in the network, determining an address of a node having a highest address in the network;

determining, based at least in part on the highest address in the network, a query timeout period;

if the node does have the lowest address in the network, listening for responses to the query from other nodes in the network and preparing a list of responding nodes; and

transferring the list of responding nodes to the caller node.

17. **(Canceled)**

18. **(Previously Presented)** A method as recited in claim 16, wherein the query from the caller node comprises a subnet mask, the node determining whether it has the lowest address in a subnet by referring to the subnet mask.

19. **(Original)** A method as recited in claim 18, wherein the address of the node having the highest address in the subnet is determined by referring to the subnet mask.

20. **(Previously Presented)** A method as recited in claim 19, wherein a query timeout period is calculated by the node by multiplying the highest address in the subnet by the delay constant.

21. **(Previously Presented)** A method of identifying a plurality of nodes on a network, comprising:

receiving a query sent from a caller node, wherein:

the query comprises a delay constant; and

the query is received by at least one of a plurality of nodes on a network;

calculating a delay period based at least in part on the delay constant and a network address associated with the at least one node;

after the delay period, transmitting an answer to the query from the at least one node;

and

monitoring, at a responder node which received the query, responses from other nodes to the query and maintaining a list of nodes which responded to the query.

22. **(Previously Presented)** The method of claim 21, wherein each node which received the query waits a respective delay period unique to the node before responding to the query.

23. **(Original)** The method of claim 21 further comprising transmitting from the responder node to the caller node after a query timeout period the list of nodes which responded to the query.

24. **(Original)** The method of claim 23, wherein the responder node is the first node to respond to the query.

25. **(Original)** The method of claim 23, wherein a selected one of the plurality of nodes is designated within the query to maintain and transmit to the caller node the list of nodes which responded to the query.

26. **(Previously Presented)** A network, comprising:

a server for posing a query to a plurality of client nodes on the network, the query comprising a delay constant; and

a plurality of client nodes for receiving the query posed by a caller node and for determining an answer to the query, each of the plurality of client nodes forwarding, after a delay period based at least in part on the delay constant and a network address, the answer to the query to the caller node, wherein:

the server receives the answers to the query from the plurality of client nodes;

and

the server maintains a list of client nodes which responded to the query.

27. **(Previously Presented)** A network as recited in claim 26, wherein each of the plurality of client nodes calculates a respective delay period to wait before forwarding its respective answer to the query to the server.

28. **(Original)** A network as recited in claim 27, wherein each of the plurality of client nodes on the network forwards the answer to the query to the server at different times.

29. **(Previously Presented)** A network as recited in claim 27, wherein each of the plurality of client nodes on the network calculates a respective delay period to wait before forwarding its respective answer to the query by multiplying the delay constant by its own network address.

30. **(Previously Presented)** A network as recited in claim 26, wherein each of the plurality of client nodes on the network are on a subnet, the query posed by the caller node comprising a subnet mask.

31. **(Previously Presented)** A system for identifying nodes on a network, comprising:

a server for sending a query to a subnet, the query comprising a delay constant;

a plurality of client nodes on the subnet for receiving the query from the server, wherein, after a delay period based at least in part on the delay constant and a network address, at least one client node on the subnet sends a local response to the query to at least one other client node on the subnet, the at least one other client node on the subnet compiling a list identifying responding nodes, the at least one other client node sending the list identifying the responding nodes to the server.

32. **(Original)** A system as recited in claim 31, wherein each of the plurality of client nodes on the subnet sends its local response at different times.

33. **(Previously Presented)** A system as recited in claim 32, wherein each of the plurality of client nodes on the network calculates a respective delay period to wait prior to sending its local response.

34. **(Original)** A system as recited in claim 32, wherein the at least one other client node on the subnet compiling the list identifying the responding nodes is a node first to respond with a local response to the query.

35. **(Original)** A system as recited in claim 34, wherein the node compiling the list identifying the responding client nodes calculates a timeout period indicating when a last of the plurality of client nodes will send its local response and receives the local responses until the timeout period has expired.

36. **(Previously Presented)** A system as recited in claim 34, wherein the timeout period is calculated by the node compiling the list identifying the responding client nodes by multiplying by the delay constant an address of a node having a highest IP address on the subnet.

37. **(Canceled)**

38. **(Previously Presented)** A system as recited in claim 36, wherein the query further comprises a subnet mask.

39. **(Previously Presented)** A system as recited in claim 31, wherein the query from the server comprises information identifying which of the plurality of client nodes on the subnet is to compile the list of responding nodes.

40. **(Original)** A system as recited in claim 39, wherein each of the responding client nodes sends its local response to the client node identified in the query.

41. **(Previously Presented)** A node for use on a subnet, comprising:

 a section for receiving a query from a caller node, the query comprising a delay constant;

 a section for determining an answer to the query;

 a section for calculating a delay period to wait before responding to the query, the delay period based at least in part on the delay constant and a network address;

 a section for determining whether the node has a lowest address in the network, wherein:

 if the node does not have the lowest address in the network, the node responds to the query after the delay period;

 if the node does have the lowest address in the network, a section of the node determines an address of a node having a highest address in the network and based at least in part on the highest address in the network, determines a query timeout period; and

 if the node does have the lowest address in the network, a section listens for responses to the query from other nodes in the network and prepares a list of responding nodes;

 and

 a section for transferring the list of responding nodes to the caller node.

42. **(Previously Presented)** A node as recited in claim 41, wherein the node determines the period to wait before responding by multiplying its network address by the delay constant.

43. **(Previously Presented)** A node as recited in claim 41, wherein the query from the caller node comprises a subnet mask, the node determining whether it has the lowest address in a subnet by referring to the subnet mask.

44. **(Original)** A node as recited in claim 43, wherein the address of the node having the highest address in the subnet is determined by referring to the subnet mask.

45. **(Previously Presented)** A node as recited in claim 44, wherein a query timeout period is calculated by the node by multiplying the highest address in the subnet by the delay constant.

46. **(Previously Presented)** A computer readable medium having computer executable code for identifying nodes on a network, comprising:

server code for posing a query to a plurality of client nodes on the network, the query comprising a delay constant; and

client code for use by a plurality of client nodes for receiving the query posed by a caller node and for determining an answer to the query, the client code comprising code instructing at least one of the plurality of client nodes to forward, after a delay period based at least in part on the delay constant and a network address, the answer to the query to the caller node, wherein a node running the server code maintains a list of client nodes which responded to the query.

47. **(Previously Presented)** A computer readable medium having computer executable code for identifying nodes on a network, comprising:

server code for use by a server for sending a query to a subnet, the query comprising a delay constant;

client code for use by a plurality of client nodes on the subnet for receiving the query from the server, wherein in response to the query, the client code for at least one of the plurality of client nodes on the subnet sends, after a delay period based at least in part on the delay constant and a network address, a local response to the query to at least one other client node on the subnet, the client code of the at least one other client node on the subnet compiling a list identifying responding nodes and sending the list identifying the responding nodes to the server.

48. **(Previously Presented)** A computer readable medium comprising computer executable code to be executed by a node on a subnet, comprising:

code for receiving a query from a caller node, the query comprising a delay constant;

code for determining an answer to the query;

code for calculating a delay period to wait before responding to the query, the delay period based at least in part on the delay constant and a network address;

code for determining whether the node has a lowest address in the network, wherein:

if the node does not have the lowest address in the network, the code directs the node to respond to the query after the delay period;

if the node does have the lowest address in the network, the code directs the node to determine an address of a node having a highest address in the network and, based at least in part on the highest address in the network, determine a query timeout period; and

if the node does have the lowest address in the network, the code directs the node to listen for responses to the query from other nodes in the network and to prepare a list of responding nodes;

and

code for transferring the list of responding nodes to the caller node.

49. **(Previously Presented)** A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to identify a plurality of nodes on a network, comprising:

instructions to receive a query posed by a caller node, wherein:

the query comprises a delay constant; and

the query is received by at least one of the plurality of nodes on the network;

instructions to calculate a delay period based at least in part on the delay constant and a network address associated with the at least one of the plurality of nodes;

instructions to transmit, after the delay period, an answer to the query from the at least one of the plurality of nodes; and

instructions to monitor, at a responder node which received the query, responses from other nodes to the query and to maintain a list of nodes which responded to the query.

50. **(Previously Presented)** A computer data signal embodied in a transmission medium comprising:

a first segment comprising receive query code to receive a query sent by a caller node to a plurality of nodes on a network, the query comprising a delay constant;

a second segment comprising delay calculating code to calculate a delay period based at least in part on the delay constant and a network address, the delay period usable to determine when to transmit a response to the query; and

a third segment comprising list processing code to monitor responses from other nodes to the query and maintain a list of nodes which responded to the query.

51. **(Previously Presented)** A network device comprising:

a receiving query section for receiving a query sent by a caller node to a plurality of nodes on a network, the query comprising a delay constant;

a delay calculating section for calculating a delay period based at least in part on the delay constant and a network address, the delay period usable to determine when to transmit a response to the query; and

a list processing section for monitoring responses from other nodes to the query and maintaining a list of nodes which responded to the query.

52. **(Previously Presented)** A network of nodes comprising:

a caller node for sending a Dynamic Query to a plurality of nodes, the Dynamic Query comprising a delay constant; and

at least one responder node comprising:

a receiving query section for receiving the Distributed Query sent by the caller node;

a delay calculating section for calculating a delay period based at least in part on the delay constant and a network address, the delay period usable to determine when to transmit a response to the query; and

a list processing section for monitoring responses from other nodes to the Distributed Query and maintaining a list of nodes which responded to the Distributed Query.